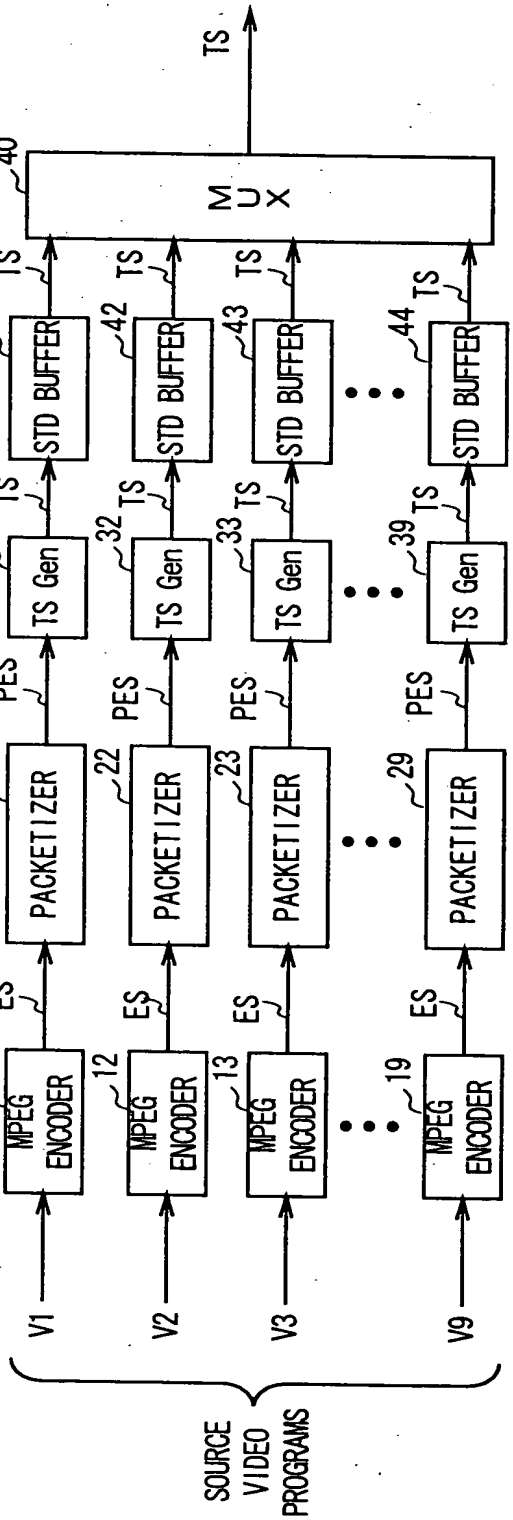

SOURCE VIDEO PROGRAMS
V1
V2
V3
V9
MPEG ENCODER
11
12
13
19
ES
PACKETIZER
21
22
23
29
PES
TS Gen
31
32
33
39
TS
STD BUFFER
41
42
43
44
MUX
40
TS

FIG. 1

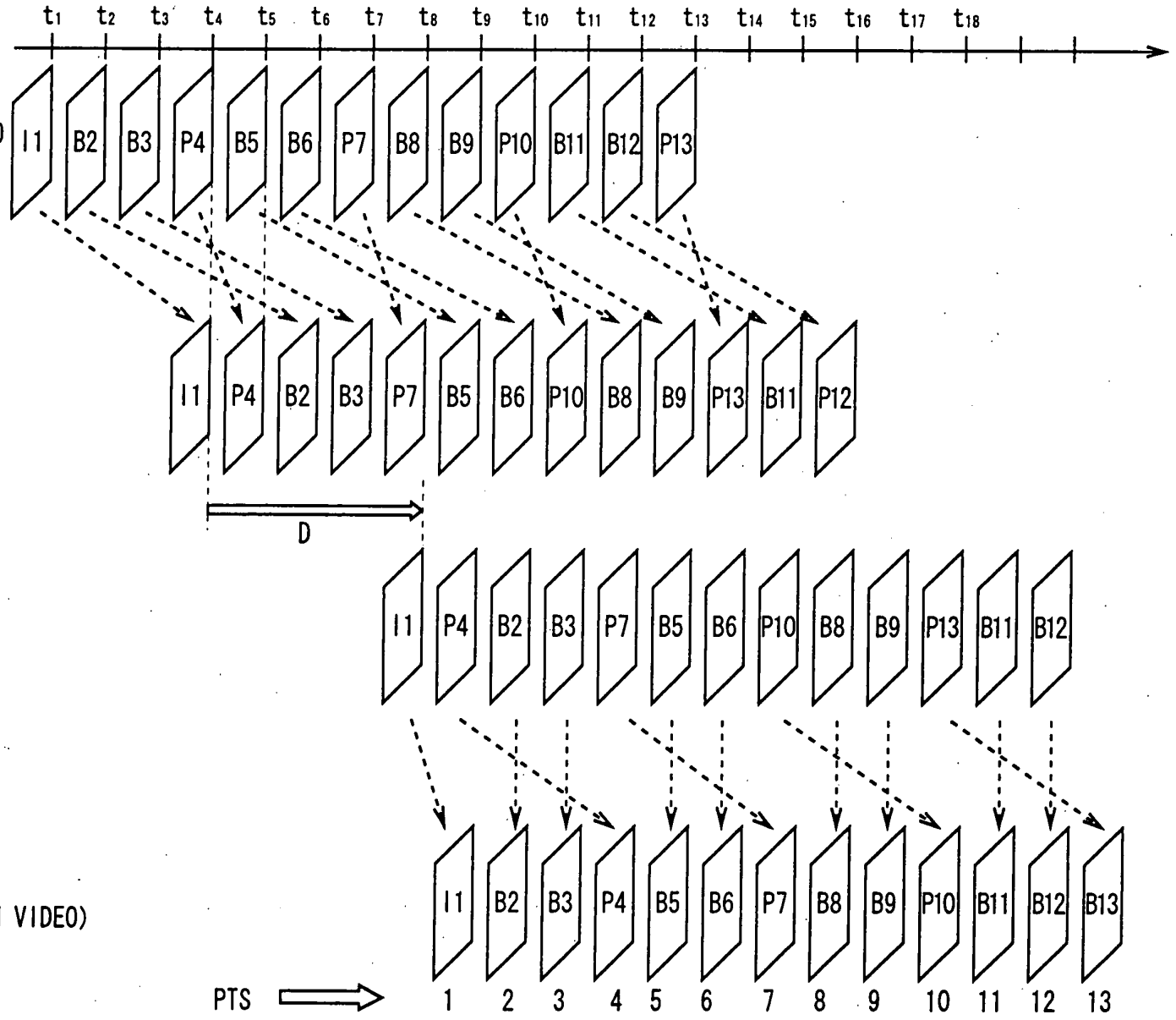


FIG. 2A

FIG. 2B

FIG. 2C

FIG. 2D

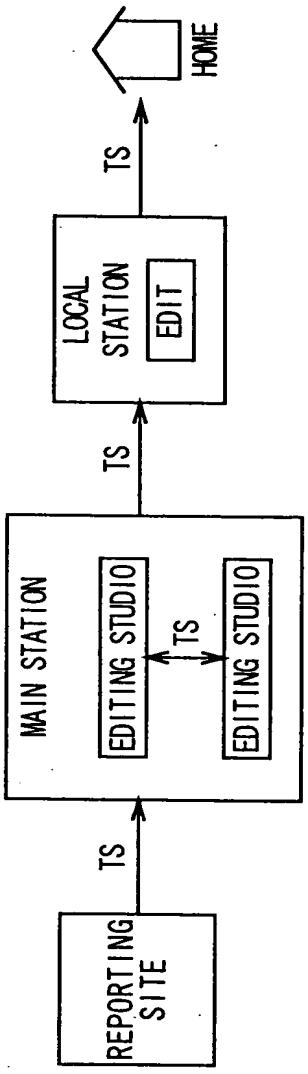

REPORTING SITE
TS
MAIN STATION
EDITING STUDIO
TS
EDITING STUDIO
TS
LOCAL STATION
EDIT
TS
HOME

FIG. 3

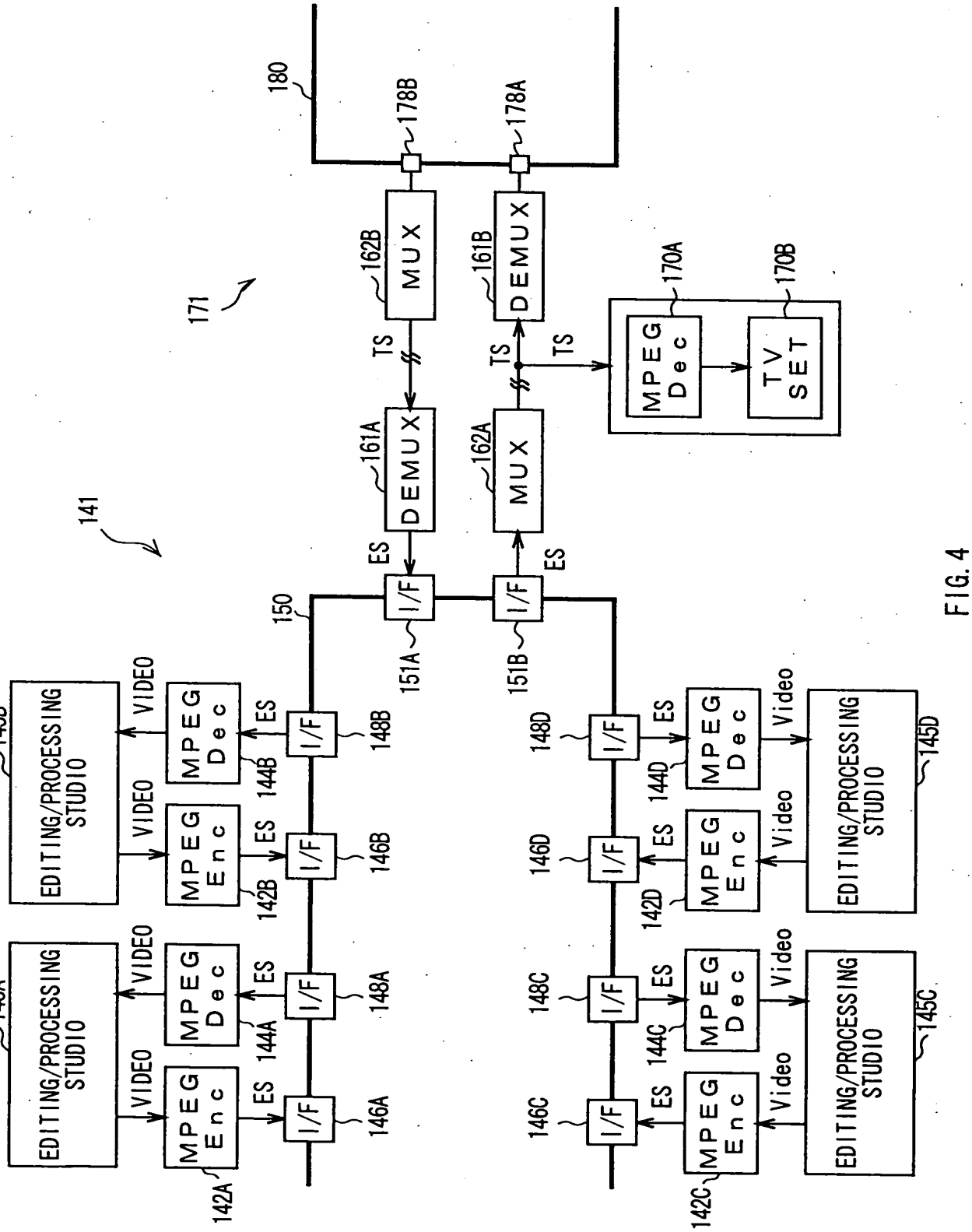

180
178B
178A
171
162B
MUX
161B
DEMUX
TS
TS
TS
161A
DEMUX
162A
MUX
MPEG Dec
170A
TV SET
170B
141
ES
ES
150
I/F
151A
I/F
151B
145B
EDITING/PROCESSING STUDIO
VIDEO
VIDEO
MPEG Enc
MPEG Dec
ES
ES
142B
144B
I/F
I/F
148B
146B
145A
EDITING/PROCESSING STUDIO
VIDEO
VIDEO
MPEG Enc
MPEG Dec
ES
ES
142A
144A
I/F
I/F
146A
148A
148D
I/F
ES
MPEG Dec
Video
144D
146D
I/F
ES
MPEG Enc
Video
142D
EDITING/PROCESSING STUDIO
145D
148C
I/F
ES
MPEG Dec
Video
144C
146C
I/F
ES
MPEG Enc
Video
142C
EDITING/PROCESSING STUDIO
145C

FIG. 4

FIG. 5A

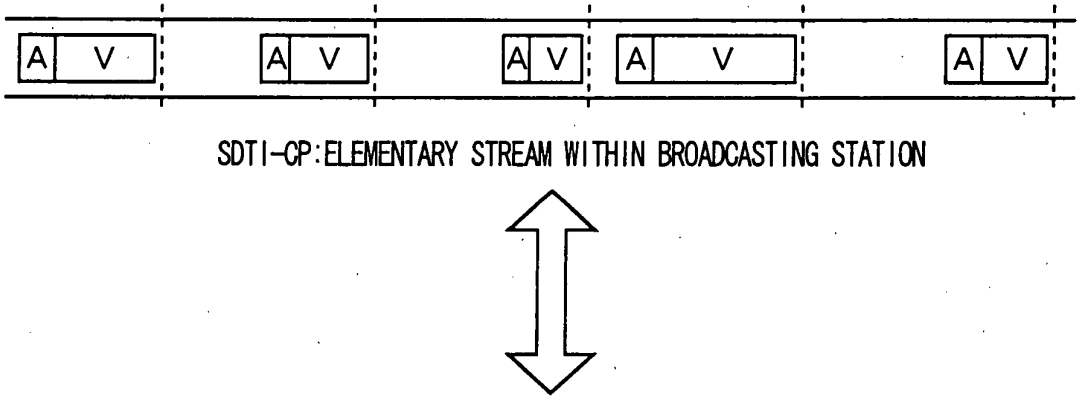

A V
A V
A V
A V
A V
SDTI-CP:ELEMENTARY STREAM WITHIN BROADCASTING STATION
MPEG:TRANSPORT STREAM BETWEEN BROADCASTING STATIONS

FIG. 5B

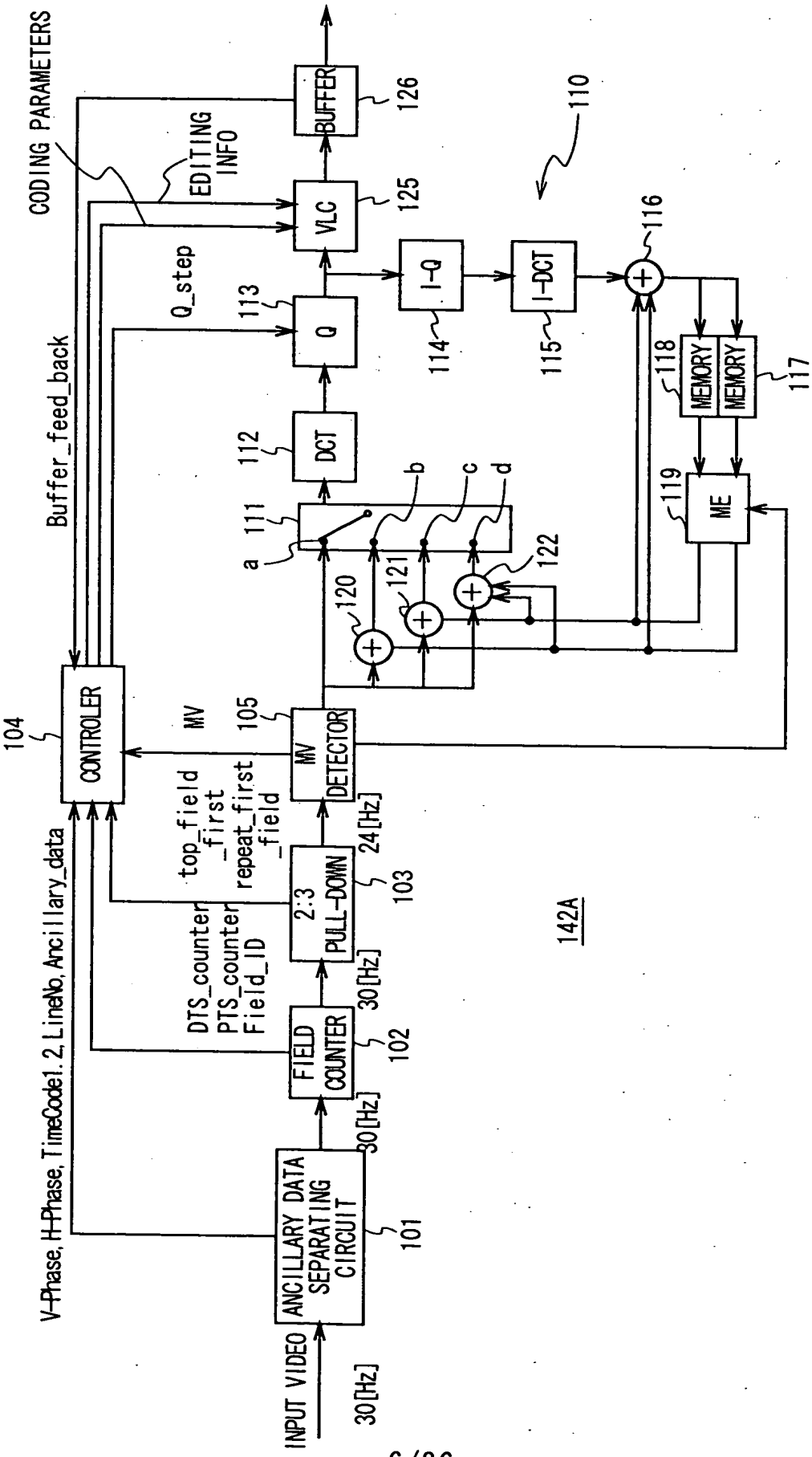

INPUT VIDEO
30[Hz]
ANCILLARY DATA SEPARATING CIRCUIT
101
FIELD COUNTER
102
2:3 PULL-DOWN
103
CONTROLER
104
MV DETECTOR
105
DCT
112
Q
113
I-Q
114
I-DCT
115
VLC
125
BUFFER
126
MEMORY
117
ME
119
CODING PARAMETERS
EDITING INFO
Buffer_feed_back
Q_step
MV
V-Phase, H-Phase, TimeCode1, 2, LineNo, Ancillary_data
DTS_counter
PTS_counter
Field_ID
top_field_first
repeat_first_field
24[Hz]
30[Hz]
142A
110
116
118
120
121
122
a
b
c
d
111

FIG. 6

FIG. 7A

VIDEO DATA SUBJECTED TO 3:2 PULL-DOWN 30[Hz]

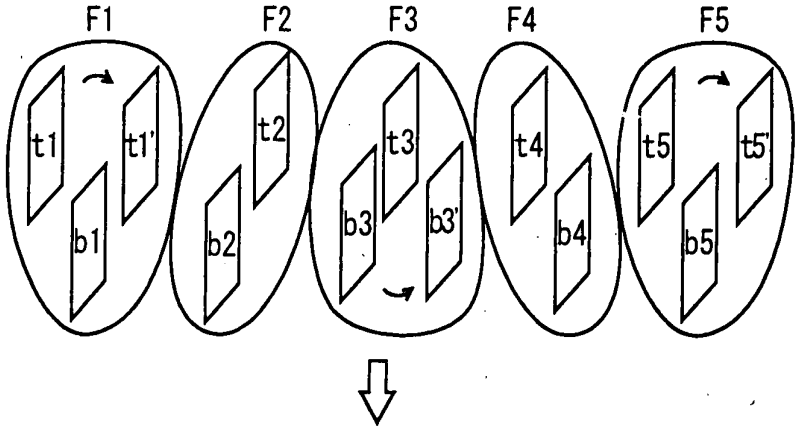


FIG. 7B

VIDEO DATA SUBJECTED TO 2:3 PULL-DOWN

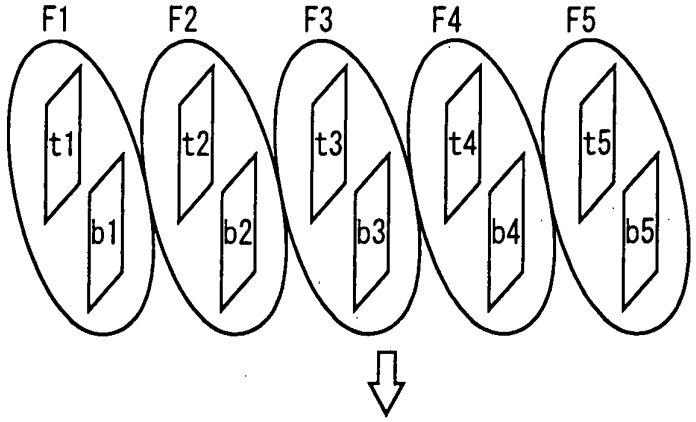
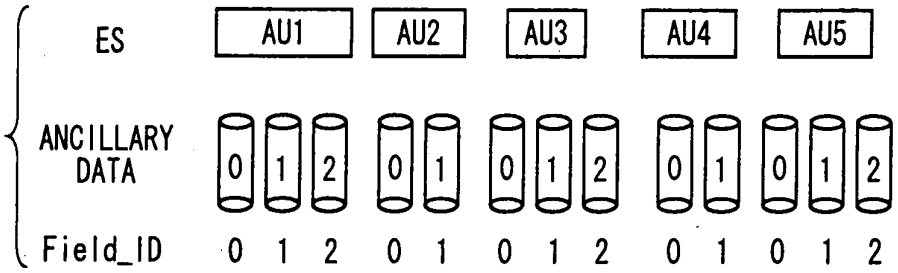


FIG. 7C

ENCODED STREAM



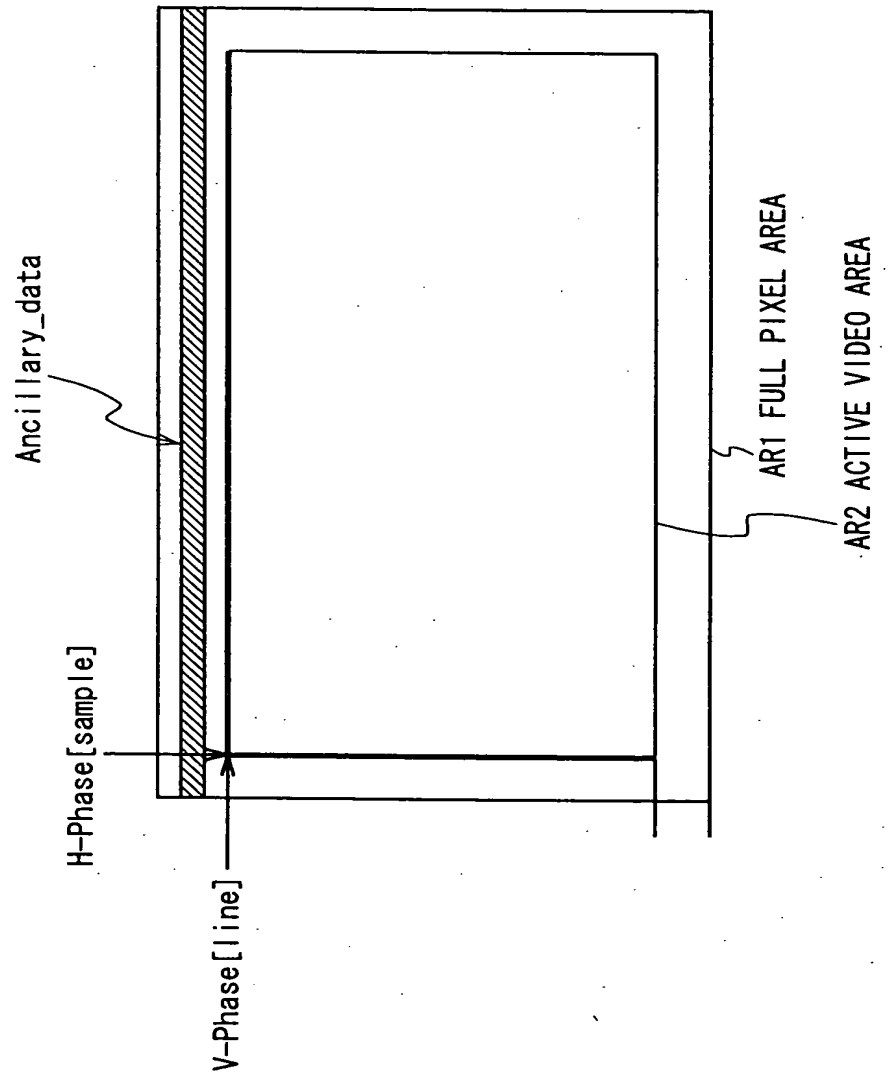

Ancillary_data
H-Phase[sample]
V-Phase[line]
AR1 FULL PIXEL AREA
AR2 ACTIVE VIDEO AREA

FIG.8

FIG. 9A

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
FRAME NO	1	2	3	4	5	6	7	8	9	10	11	12	13
PICTURE TYPE	I	B	B	P	B	B	P	B	B	I	B	B	P
Repeat_first_field	1	0	1	0	1	0	1	0	1	0	1	0	1
Top_field_first	1	0	0	1	1	0	0	1	1	0	0	1	1
FRAME STRUCTURE													

FIG. 9B

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
FRAME NO	1	2	3	4	5	6	7	8	9	10	11	12	13
PICTURE TYPE	I	B	B	P	B	B	P	B	B	I	B	B	P
Repeat_first_field	1	0	1	0	1	0	1	0	1	0	1	0	1
Top_field_first	1	0	0	1	1	0	0	1	1	0	0	1	1
FRAME STRUCTURE													
PTS_counter	0	3	5	8	10	13	15	18	20	23	25	28	30

FIG. 9C

	F1	F4	F2	F3	F7	F5	F6	F10	F8	F9	F13	F11	F12
FRAME NO	1	4	2	3	7	5	6	10	8	9	13	11	12
PICTURE TYPE	I	P	B	B	P	B	B	I	B	B	P	B	B
Repeat_first_field	1	0	0	1	1	1	0	0	0	1	1	1	0
Top_field_first	1	1	0	0	0	1	0	0	1	1	1	0	1
FRAME STRUCTURE													
DTS_counter	125	0	3	5	8	10	13	15	18	20	23	25	28

video_sequence(){	No. of bits	Mnemonic
next_start_code()		
sequence_header()		
sequence_extension()		
do{		
extension_and_user_data(0)		
do{		
if(nextbits()==group_start_code){		
group_of_pictures_header()		
extension_and_user_data(1)		
}		
picture_header()		
picture_coding_extension()		
extension_and_user_data(2)		
picture_data()		
}while((nextbits()==picture_start_code) |		
(nextbits()==group_start_code))		
if(nextbits()!=sequence_end_code){		
sequence_header()		
sequence_extension()		
}		
}while(nextbits()!=sequence_end_code)		
sequence_end_code	32	bslbf
)		

FIG. 10

sequence_header(){	No. of bits	Mnemonic
sequence_header_code	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	"1"
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	
load_intra_quantiser_matrix	1	
if(load_intra_quantiser_matrix)		
Intra_quantiser_matrix [64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix [64]	8*64	uimsbf
next_start_code()		
}		

FIG. 11

picture_data(){	No. of bits	Mnemonic
do{		
slice()		
}while(nextbits()==slice_start_code)		
next_start_code()		
}		

FIG. 25

sequence_extention(){	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	uimsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	unimsbf
frame_rate_extension_d	5	uimsbf
next_start_code()		
}		

Mnemonic

FIG. 12

extension_and_user_data(i){	No. of bits	Mnemonic
while((ll=1)lts()&&(nextbits()=extension_start_oode)) l		
(nextblts()--user_data_start_code)){		
if(nextblts()--extenslon_start_code)		
extension_data(i)		
if(nextbits()=user_data_start_code)		
user_data()		
}		
}		

FIG. 13

Syntax	Bits	Mnemonic
user_data() {		
user_data_start_code	32	
MPEG_ES_Editing_Information(i) {		
if(i—0) {		
while(nextbits() != "0000 0000 0000 0000 0000 0001") {		
if(nextbits()== "V-phase")		
V-phase()		
Else if(nextbits()== "H-phase")		
H-phase()		
{		
{		
Else if(i==2) {		
while(nextbits() != "0000 0000 0000 0000 0000 0001") {		
if((nextbits()== "Time code1") | (nextbits()== "Time code2"))		
Time_code()		
Else if(nextbits()== "Picture Order")		
Picture_order()		
{		
while(nextbits() != "0000 0000 0000 0000 0000 0001") {		
if(nextbits()== "Ancillary_data")		
Ancillary_data()		
Else if(nextbits()== "History data")		
History data()		
{		
{		
if(nextbits()== "User data")		
User_data()		
Next_start_code()		
}		

FIG. 14

Data_ID	Data_type
00	FORBIDDEN
01	V–Phase
02	H–Phase
03	Time code 1
04	Time code 2
05	Picture Order
06	Video Index
07	Ancillary data
08	History data
...	...
80	Control flags
...	...
FF	User data

FIG. 15

Syntax	Bits	Mnemonic
V–Phase(){		
Data_ID	8	bslbf
V–Phase	16	uimsbf
}		

FIG. 16

Syntax	Bits	Mnemonic
H−Phase(){		
Data_ID	8	bslbf
H−Phase	8	uimsbf
}		

FIG. 17

Syntax	Bits	Mnemonic
Picture_order(){		
Data_ID	8	bslbf
DTS_presence	1	bslbf
PTS_counter	7	uimsbf
If(DTS_presence-- "1"){		
Marker_bits	1	bslbf
DTS_counter	7	uimsbf
}		
}		

FIG. 20

Syntax	Bits	Mnemonic
Time_code(){		
Data_ID	8	bslbf
Time_code [63..48]	16	uimsbf
Marker_bit	1	bslbf
Time_code [47..32]	16	uimsbf
Marker_bit	1	bslbf
Time_code [31..16]	16	uimsbf
Marker_bit	1	bslbf
Time_code [15..0]	16	uimsbf
Marker_bit	1	bslbf
Reserved_bits	4	bslbf
}		

FIG. 18

Syntax	Bits	Mnemonic
Time_code [63..0]{		
Color frame flag	1	bslbf
Drop frame flag (NTSC)/unused (PAL)	1	bslbf
TV frame tens	2	uimsbf
TV frame units	4	uimsbf
Field phase (NTSC)/binary group flag 0 (PAL)	1	bslbf
TV seconds tens	3	uimsbf
TV seconds units	4	uimsbf
Binary group flag 0 (NTSC)/binary group flag 2 (PAL)	1	bslbf
TV minutes tens	3	uimsbf
TV minutes units	4	uimsbf
Binary group flag 2 (NTSC)/field phase (PAL)	1	bslbf
Binary group flag 1 (NTSC)/binary group flag 1 (PAL)	1	bslbf
TV hours tens	2	uimsbf
TV hours units	4	uimsbf
2nd binary group	4	uimsbf
1st binary group	4	uimsbf
4th binary group	4	uimsbf
3rd binary group	4	uimsbf
6th binary group	4	uimsbf
5th binary group	4	uimsbf
8th binary group	4	uimsbf
7th binary group	4	uimsbf
}	4	uimsbf

FIG. 19

Syntax	Bits	Mnemonic
Ancillary_data(){		
Data_ID	8	bslbf
Field_ID	2	bslbf
Line_number	14	uimsbf
Ancillary_data_length	16	uimsbf
Marker_bits	1	bslbf
For(j=0; j<Ancillary_data_length; j++){		
Ancillary_data_payload	22	uimsbf
Marker_bits	1	bslbf
}		
While(!bytealigned())		
Zero_bit	1	"0"
}		

FIG. 21

group_of_picture_header(){	No. of bits	Mnemonic
group_start_code	32	bslbf
time_code	25	bslbf
closed_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code()		
}		

FIG. 22

picture_header(){	No. of bits	Mnemonic
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type==2 || picture_coding_type==3){		
full_pel_forward_vector	1	
forward_f_code	3	uimsbf
}		
if(picture_coding_type==3){		
full_pel_backward_vector	1	
backward_f_code	3	uimsbf
}		
while(nextbits()== "1"){		
extra_bit_picture/*with the value "1" */	1	uimsbf
extra_information_picture	8	
}		
extra_bit_picture/*with the value "0" */	1	uimsbf
next_start_code()		
}		

FIG. 23

picture_coding_extension(){	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code [0][0]/*forward horizontal*/	4	uimsbf
f_code [0][1]/*forward vertical*/	4	uimsbf
f_code [1][0]/*backward horizontal*/	4	uimsbf
f_code [1][1]/*backward vertical*/	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf
top_field_first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if(composite_display_flag){		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code()		
}		

FIG. 24

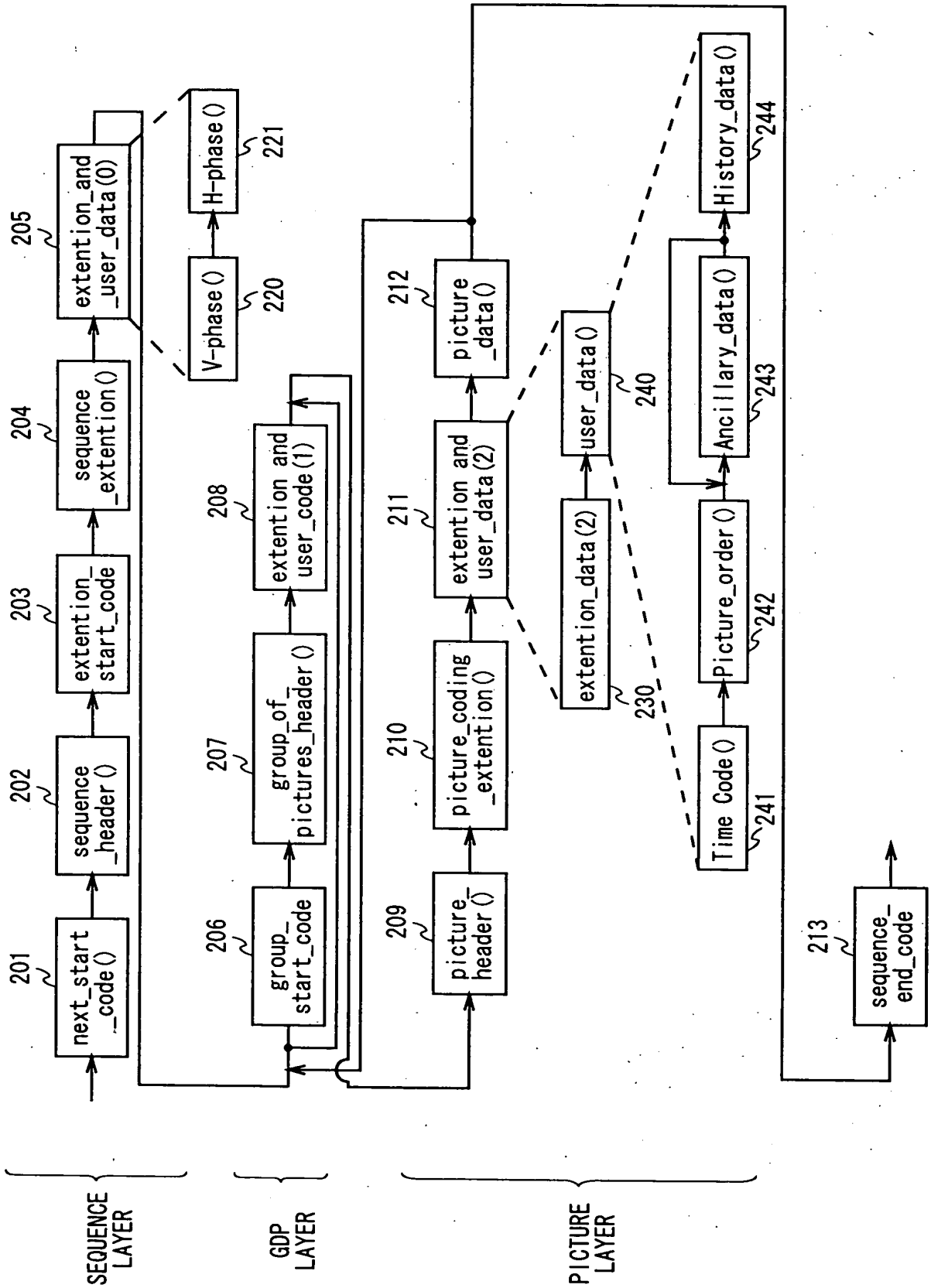

SEQUENCE LAYER
GDP LAYER
PICTURE LAYER
201 next_start_code()
202 sequence_header()
203 extention_start_code
204 sequence_extention()
205 extention_and_user_data(0)
206 group_start_code
207 group_of_pictures_header()
208 extention and user_code(1)
209 picture_header()
210 picture_coding_extention()
211 extention and user_data(2)
212 picture_data()
213 sequence_end_code
220 V-phase()
221 H-phase()
230 extention_data(2)
240 user_data()
241 Time Code()
242 Picture_order()
243 Ancillary_data()
244 History_data()

FIG. 26

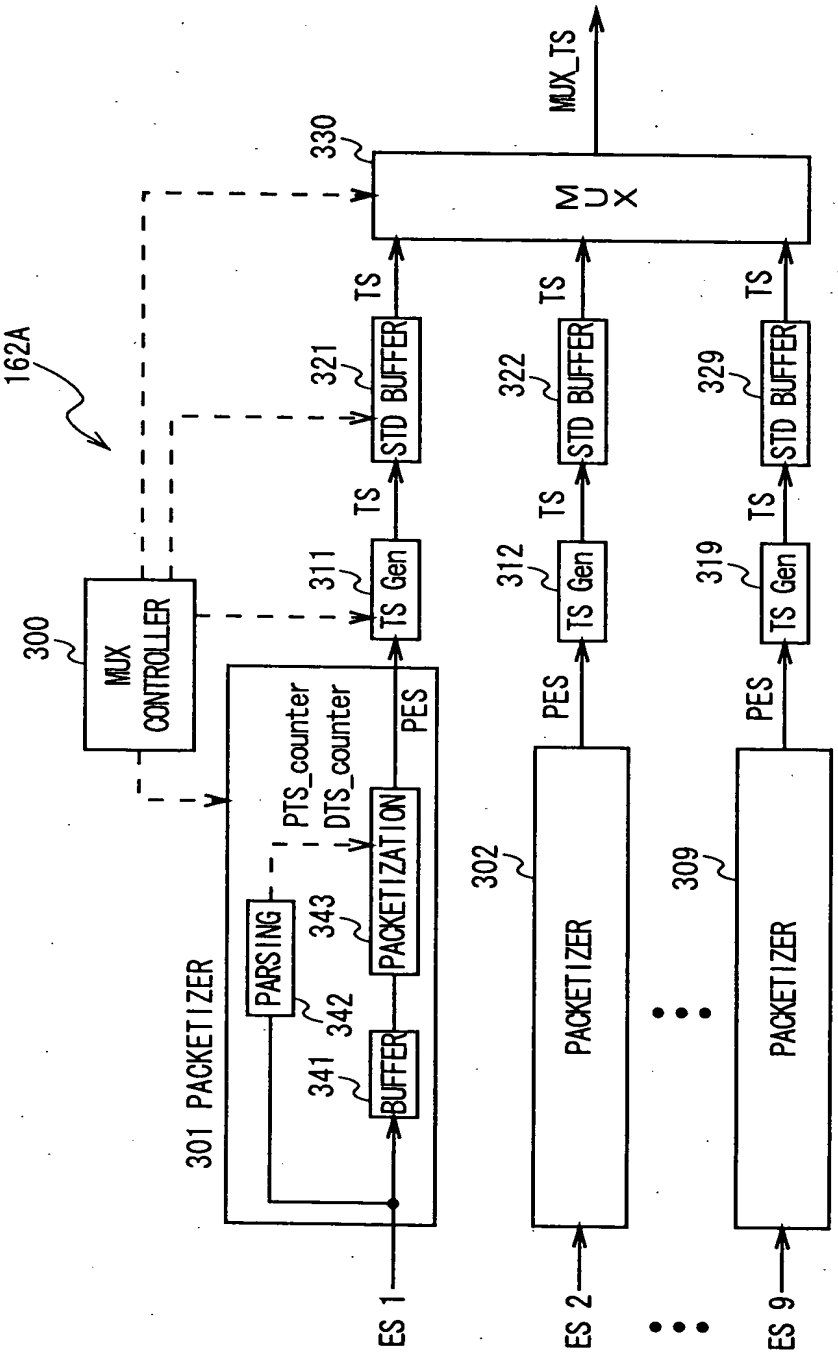

MUX_TS
330
MUX
162A
300
MUX CONTROLLER
321
322
329
STD BUFFER
TS
311
312
319
TS Gen
PES
301 PACKETIZER
PTS_counter
DTS_counter
PARSING
343
PACKETIZATION
341
342
BUFFER
302
309
PACKETIZER
ES 1
ES 2
ES 9

FIG. 27

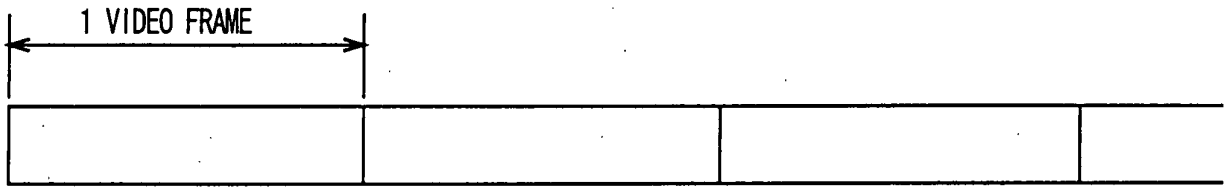


FIG. 28A SOURCE VIDEO DATA

FIG. 28B ES(ELEMENTARY STREAM)

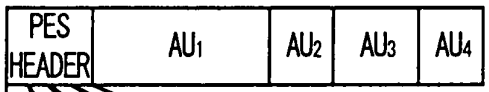


FIG. 28C PES PACKET

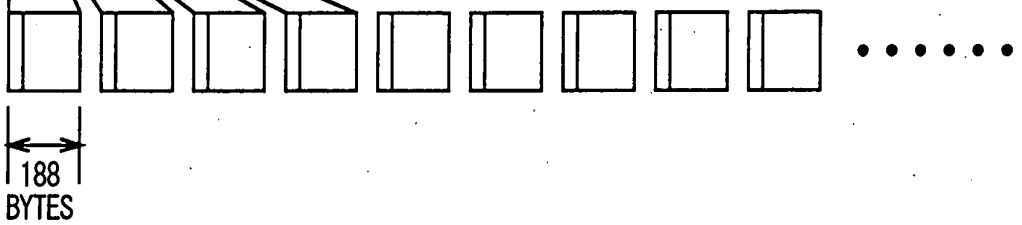


FIG. 28D TS PACKET

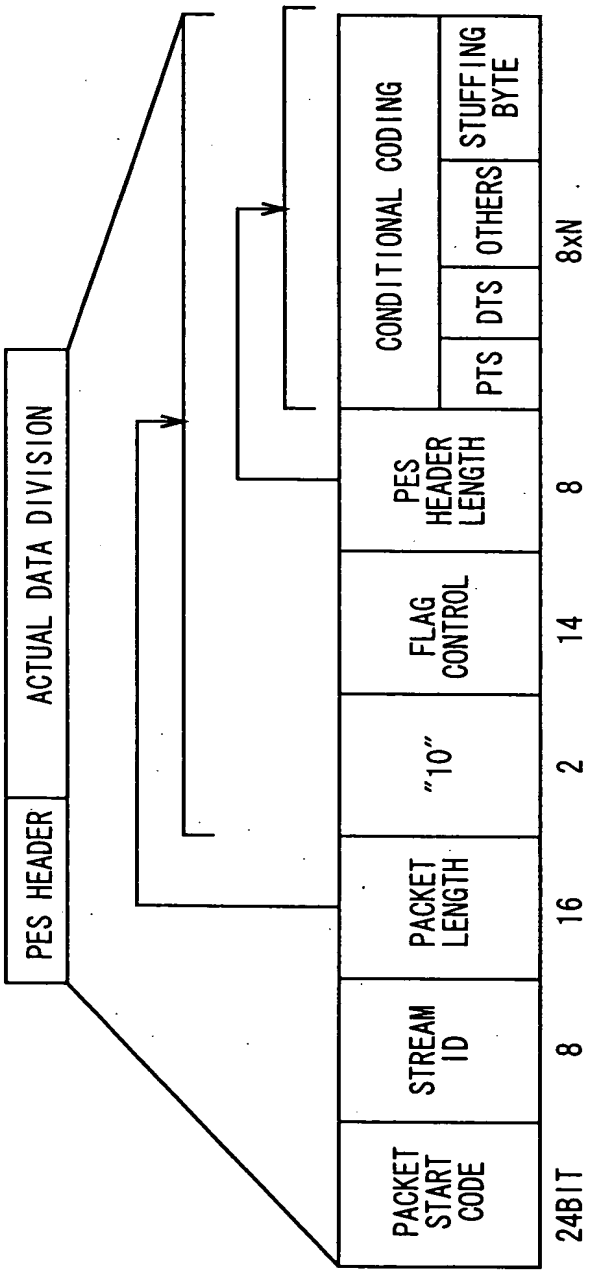
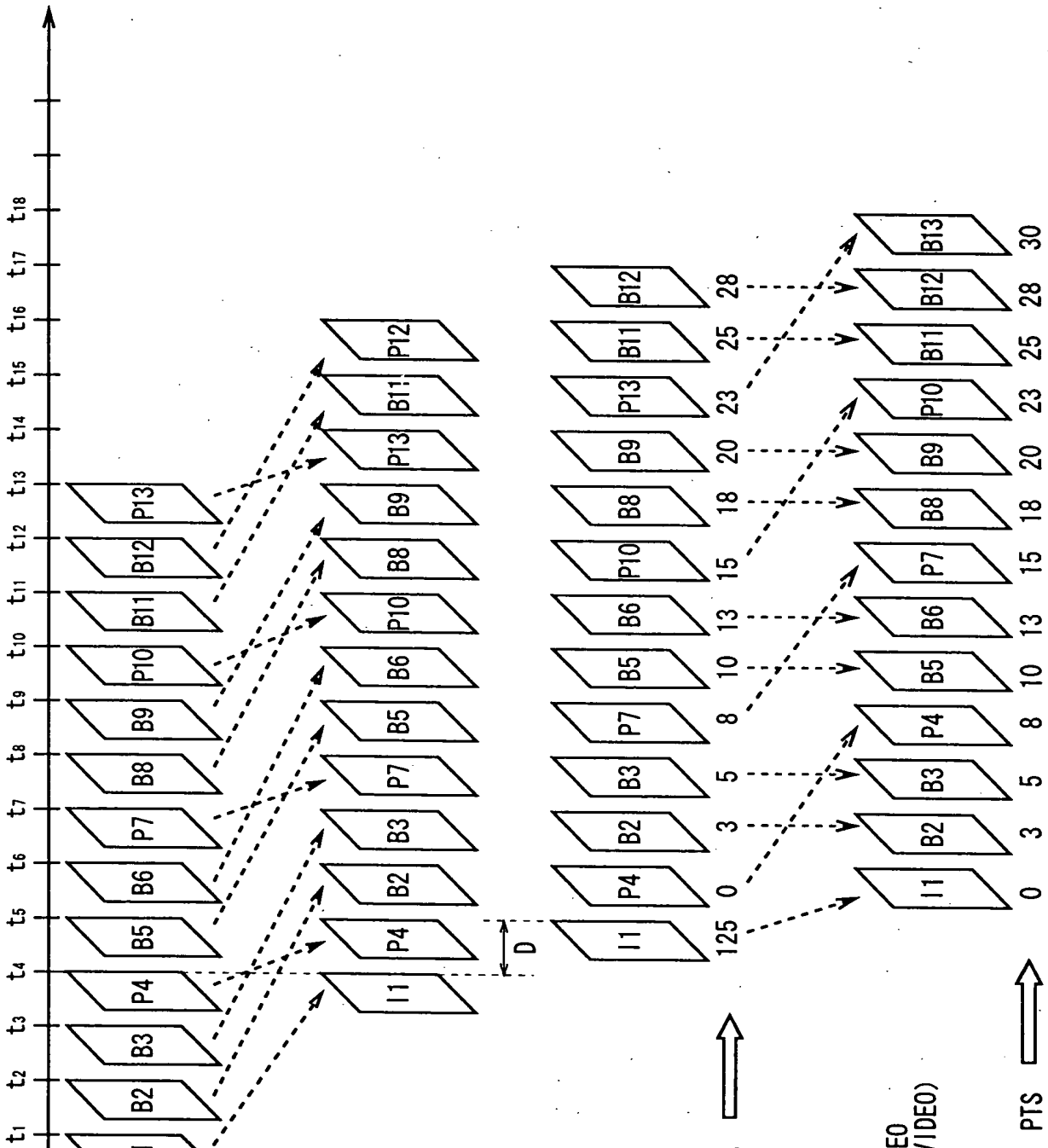

PES HEADER
ACTUAL DATA DIVISION
PACKET START CODE
STREAM ID
PACKET LENGTH
"10"
FLAG CONTROL
PES HEADER LENGTH
CONDITIONAL CODING
PTS
DTS
OTHERS
STUFFING BYTE
24BIT
8
16
2
14
8
8xN

FIG. 29



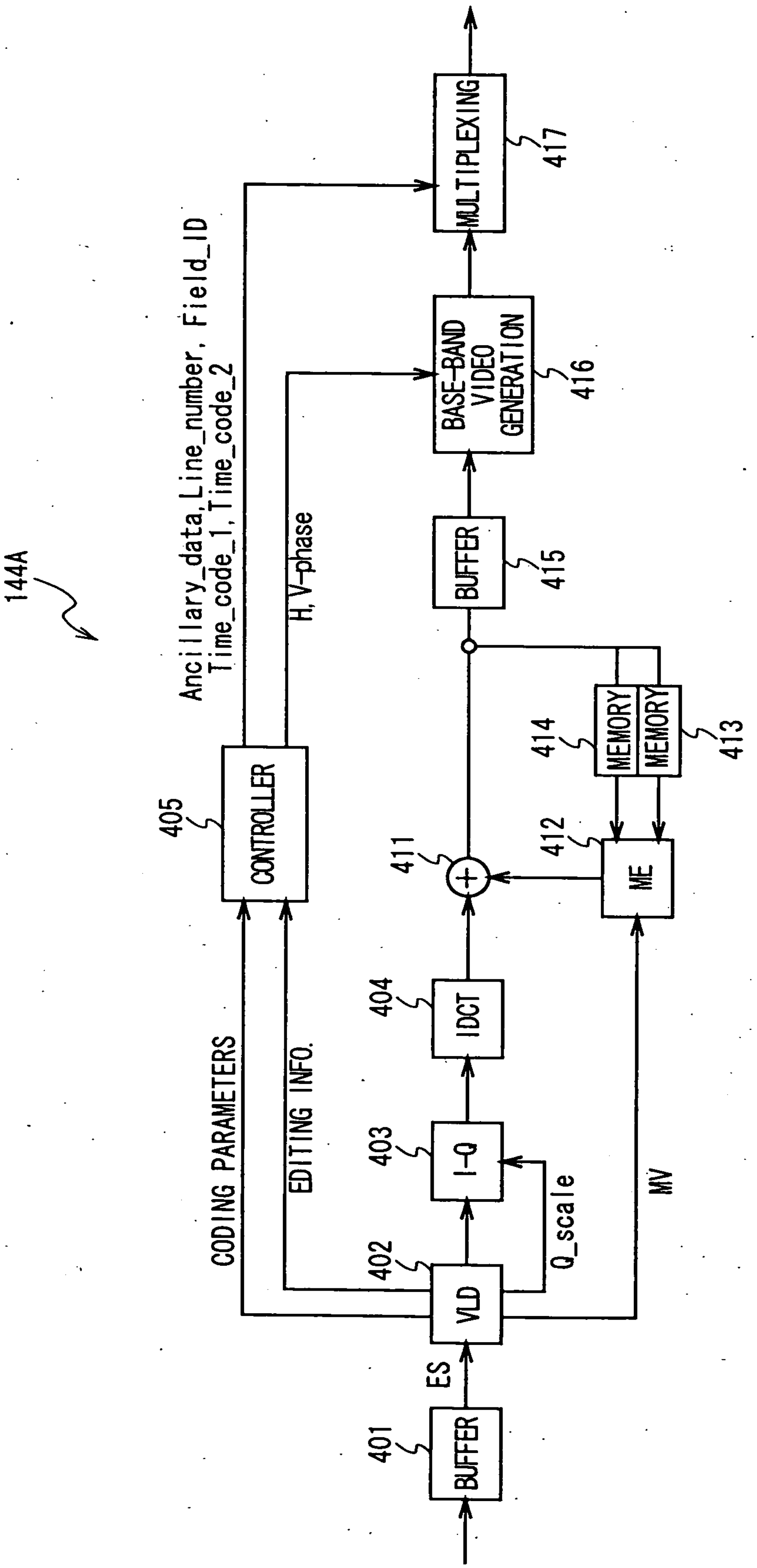

144A
BUFFER
401
ES
VLD
402
I-Q
403
IDCT
404
CONTROLLER
405
+
411
ME
412
MEMORY
413
MEMORY
414
BUFFER
415
BASE-BAND VIDEO GENERATION
416
MULTIPLEXING
417
CODING PARAMETERS
EDITING INFO.
Q_scale
MV
H, V-phase
Ancillary_data, Line_number, Field_ID
Time_code_1, Time_code_2

FIG. 31

Explanation of Reference Numerals

1 ... Video processor, 2 ... MPEG encoder, 3 ... MPEG decoder, 5 ... 3:2 pull-down process, 6 ... 2:3 pull-down process, 104 ... Controller, 105 ...motion vector detector, 112 ... DCT circuit, 113, ... Quantizing circuit, 119 ... Motion compensating circuit, 125 ... Variable-length coding circuit, 126 ... Send buffer, 142A ... MPEG encoder, 300 ... Multiplexing controller, 301, 302, 309 ... Packetizers, 330 ... Multiplexer.